Instituto de Sistemas e Robótica

Pólo de Lisboa

Rational and Convergent Model-Free Adaptive Learning for Team Markov Games[1]

Francisco S. Melo
M. Isabel Ribeiro

February 2007
RT-601-07

ISR Torre Norte
Av. Rovisco Pais, 1
1049-001 Lisboa
PORTUGAL

[1]This work was partially supported by Programa Operacional Sociedade do Conhecimento (POS_C) that includes FEDER funds. The first author acknowledges the PhD grant SFRH/BD/3074/2000.

Rational and Convergent Model-Free Adaptive Learning for Team Markov Games

Francisco A. Melo **M. Isabel Ribeiro**
Institute for Systems and Robotics
Instituto Superior Técnico
Av. Rovisco Pais, 1
1049-001 Lisboa,
PORTUGAL
{fmelo,mir}@isr.ist.utl.pt

Abstract

In this paper, we address multi-agent decision problems where all agents share a common goal. This class of problems is suitably modeled using finite-state Markov games with identical interests. We tackle the problem of coordination and contribute a new algorithm, *coordinated Q-learning* (CQL). CQL combines Q-learning with *biased adaptive play*, a coordination mechanism based on the principle of fictitious-play. We analyze how the two methods can be combined without compromising the convergence of either. We illustrate the performance of CQL in several different environments and discuss several properties of this algorithm.

Recent years have witnessed increasing interest in extending reinforcement learning (RL) to multi-agent problems. However, reinforcement learning methods often require the environment to be *stationary*. If a learning agent interacting with an environment where other agents co-exist can disregard them as part of the environment, there is an implicit violation of the stationarity assumption that can lead to poor convergence of the learning algorithms. Even if convergence is attained, the learned policy can be unsatisfactory.

Markov games (also known as stochastic games), understood as extensions of Markov processes to multi-agent scenarios, have thoroughly been used as suitable models to address multi-agent reinforcement learning problems, and several researchers adapted classical RL methods to this multi-agent framework.

Littman [16] proposed the Minimax-Q algorithm as a possible application of Q-learning to zero-sum Markov games. Hu and Wellman [12] later proposed Nash-Q, an elaboration on Minimax-Q that can be applied to general-sum Markov games. They established convergence of Nash-Q under quite stringent conditions, thus leading to the development of Friend-or-Foe Q-learning (FFQ) [18]. FFQ requires less stringent assumptions than Nash-Q, while retaining its convergence properties in several classes of Markov games.

Claus and Boutilier [7] proposed joint-action learners (JAL), combining Q-learning with fictitious play in team Markov games. Uther and Veloso [25] combined fictitious play with prioritized sweeping to address planning in adversarial scenarios. Gradient-based learning strategies are analyzed with detail in [4, 22]; Bowling and Veloso [5] propose a policy-based learning method that applies a policy hill-climbing strategy with varying step, using the principle of "win or learn fast" (WoLF-PHC). Many other works on multi-agent learning systems can be found in the literature (see, for example, the surveys [3, 20]).

In this paper, we address finite-state Markov games with identical interests (henceforth referred as *team Markov games*). We cast this class of games as a generalization of Markov decision

processes (in the line of [24]) and describe a straightforward application of Q-learning to such scenarios. We then tackle the problem of *coordination* or *equilibrium selection*. We contribute the *coordinated Q-learning algorithm* (CQL), combining Q-learning with *biased adaptive play* (BAP).[1] BAP is a sound coordination mechanism introduced in [26] and based on the principle of fictitious-play. We analyze how BAP can be interleaved with Q-learning without affecting the convergence of either method, thus establishing convergence of CQL. We also illustrate the performance of CQL in several different environments and discuss several properties of the methods (such as its convergence and rationality).

The paper is organized as follows. We start by describing the framework of Markov games as an extension of Markov decision processes and matrix games, and describe how Q-learning can be applied to this class of games. We then address the problem of equilibrium selection and introduce biased adaptive play, analyzing its main properties. We proceed with the detailed description of CQL and its convergence properties. We present several results obtained with the algorithm and conclude the paper with some discussion on the performance algorithm and future work.

1 Markov games and equilibrium selection

Markov games arise as a suitable framework to address reinforcement learning problems in multi-agent scenarios. This framework can be seen as a multi-agent extension of Markov decision processes. In this section, we describe Markov games and some important concepts that concern it.

1.1 Markov decision processes

Let $\mathcal{X}$ be a finite set of states and $\{X_t\}$ a $\mathcal{X}$-valued controlled Markov chain. The transition probabilities for the chain are given by a probability function

$$\mathbb{P}\left[X_{t+1} = j \mid X_t = i, A_t = a\right] = \mathsf{P}_a(i,j),$$

where $i, j \in \mathcal{X}$ and $a \in \mathcal{A}$. The $\mathcal{A}$-valued process $\{A_t\}$ represents the control process: A_t is the control action at time instant t and $\mathcal{A}$ is the finite set of possible actions. A decision-maker aims at choosing the control process $\{A_t\}$ so as to maximize the infinite-horizon total discounted reward

$$V(\{A_t\}, i) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(X_t, A_t) \mid X_0 = i\right],$$

where $0 \leq \gamma < 1$ is a discount-factor and $R(i,a)$ represents a random "reward" received for taking action $a \in \mathcal{A}$ in state $i \in \mathcal{X}$. We assume throughout the paper that there is a deterministic function $r : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \longrightarrow \mathbb{R}$ assigning a reward $r(i,a,j)$ every time a transition from i to j occurs after taking action a. This simplifies the notation without introducing a great loss in generality.

We refer to the tuple $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ as a *Markov decision process* (MDP). Given the MDP $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$, the *optimal value function* V^* is defined for each state $i \in \mathcal{X}$ as

$$V^*(i) = \max_{\{A_t\}} \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^t R(X_t, A_t) \mid X_0 = i\right]$$

and verifies

$$V^*(i) = \max_{a \in \mathcal{A}} \sum_{i \in \mathcal{X}} \left[r(i,a,j) + \gamma V^*(i)\right] \mathsf{P}_a(i,j),$$

[1] We must emphasize that our method, although baring a somewhat related designation, has no relation whatsoever with the *coordinated reinforcement learning* algorithms proposed in [11].

which is a form of the Bellman optimality equation. The optimal Q-values $Q^*(i,a)$ are defined for each state-action pair $(i,a) \in \mathcal{X} \times \mathcal{A}$ as

$$Q^*(i,a) = \sum_{i \in \mathcal{X}} \left[r(i,a,j) + \gamma V^*(i) \right] \mathsf{P}_a(i,j). \tag{1.1}$$

If $V^*(i)$ "measures" the total discounted reward obtained during an expectedly optimal trajectory starting at state i, $Q^*(i,a)$ measures the total discounted reward obtained during an expectedly optimal trajectory starting at state i when the first action is a.

The optimal Q-function can be approximated by a sequence of functions $\{Q_n\}$, generated recursively by

$$Q_{n+1}(i,a) = Q_n(i,a) + \alpha_n(i,a) \left[R(i,a) + \gamma \max_{b \in \mathcal{A}} Q_n(X(i,a),b) - Q_n(i,a) \right], \tag{1.2}$$

where $X(i,a)$ is random variable obtained according to the transition probabilities defined by P and $\{\alpha_n(i,a)\}$ is a sequence of step-sizes verifying $\sum_n \alpha_n(i,a) = \infty$ and $\sum_n \alpha_n^2(i,a) < \infty$. The sequence $\{Q_n\}$ will converge to Q^* as long as each pair (i,a) is "visited" infinitely often [27]. Expression (1.2) is the update equation of *Q-learning*, a widely known method that we use in our multi-agent algorithm.

1.2 Matrix games

A matrix game is a tuple $(N, \mathcal{A}, r)$, where N is a set of players, $\mathcal{A} = \times_{k=1,\ldots,N} \mathcal{A}^k$ is the set of all *joint actions* and $r = \times_{k=1,\ldots,N} r^k$ is a function assigning a utility or payoff $r(a) = (r^1(a), \ldots, r^N(a))$ to each joint action $a \in \mathcal{A}$.[2]

The game is played as follows. Each player $k \in N$ chooses an individual action a^k from its individual set of actions $\mathcal{A}^k$. Then, all N players *simultaneously* play the corresponding actions and, according to the resulting joint action a, each player receives a reward $r^k(a)$. We denote a *joint action* or *action profile* by $a = (a^k, \ldots, a^k)$ and a reduced action obtained from a by removing the individual action a^k of player k by a^{-k}.

Matrix games can usually be represented by N matrices whose elements define the individual payoffs of each player for each joint action. Figure 1 represents a possible matrix game with 2 players.

	a	b
a	5	-10
b	-10	0

Reward for player 1

	a	b
a	0	-10
b	-10	5

Reward for player 2

Figure 1: Example of a matrix game. In this game, $N = 2$ and $\mathcal{A}^k = \{a, b\}$, $k = 1, 2$.

An individual *strategy* σ^k defines the probability of player k playing each action $a^k \in \mathcal{A}^k$ in the game. Clearly, $\sum_{a^k \in \mathcal{A}^k} \sigma^k(a^k) = 1$. A strategy σ^k is a *pure strategy* if $\sigma^k(a^k) = 1$ for some action $a^k \in \mathcal{A}^k$ and *a mixed strategy* otherwise. A *strategy profile* is a vector $\sigma = (\sigma^1, \ldots, \sigma^N)$ of individual strategy profiles and $\sigma(a)$ represents the probability of playing the joint action a when all agents follow the strategy σ. We refer to σ_{-k} as a *reduced strategy profile* or simply as *reduced strategy* and is obtained from σ by removing the individual strategy σ^k of player k.

[2] We use the notation $\times_{k=1,\ldots,N} U^k$ to represent the cartesian product of the sets $U^k, k = 1, \ldots, N$.

The individual strategy $(\sigma^k)^*$ of player k is a *best response* to a reduced strategy σ^{-k} if player k cannot improve its expected reward by using any other individual strategy σ^k. Formally, this is stated as

$$\sum_a (\sigma^{-k}, (\sigma^k)^*)(a) r^k(a) \geq \sum_a (\sigma^{-k}, \sigma^k)(a) r^k(a),$$

for any individual strategy σ^k.

A *Nash equilibrium* is a strategy profile $\sigma^* = ((\sigma^*)^1, \ldots, (\sigma^*)^k)$ in which each individual strategy $(\sigma^*)^k$ is a best response for player k to the reduced strategy $(\sigma^*)^{-k}$. Every finite matrix game has at least one Nash equilibrium. In the example of Figure 1, the pure strategies (a, a) and (b, b) are Nash equilibria.

Matrix games can be classified according to their utility function as *zero-sum games*, where $N = 2$ and $r^1(a) = -r^2(a)$ for all $a \in \mathcal{A}$ and *general-sum games*, otherwise. General sum games include *team games* as a particular case. In a team game, $r^1(a) = \ldots = r^N(a)$ for all $a \in \mathcal{A}$. Zero-sum Markov games are also known as *fully competitive* and team games as *fully cooperative*.

1.3 Coordination and Equilibrium selection

When considering finite, fully cooperative games, the method of *fictitious play* [6] is known to converge to a Nash equilibrium [19]. However, if there are multiple such equilibria with different values, there are no guarantees that the attained equilibrium is the one with highest value.

Consider, for example, the fully cooperative game in Fig. 2.

	$\beta_a\gamma_a$	$\beta_a\gamma_b$	$\beta_a\gamma_c$	$\beta_b\gamma_a$	$\beta_b\gamma_b$	$\beta_b\gamma_c$	$\beta_c\gamma_a$	$\beta_c\gamma_b$	$\beta_c\gamma_c$
α_a	**10**	-20	-20	-20	-20	5	-20	5	-20
α_b	-20	-20	5	-20	**10**	-20	5	-20	-20
α_c	-20	5	-20	5	-20	-20	-20	-20	**10**

Figure 2: Fully cooperative game with multiple equilibria.

In this game, three players α, β and γ repeatedly engage in the matrix game described by the reward table in Fig. 2 and receive the same reward. Each player has 3 available actions, a, b and c. In Fig. 2 we represent by x_y action y of player x, e.g., α_a is action a of player α. The boldface entries in Figure 2 represent optimal equilibria, (a, a, a), (b, b, b) and (c, c, c). Notice that only if the whole team coordinates in playing one same optimal equilibrium does each player receive the maximum payoff. For example, if players α and β decide to play the equilibrium (a, a, a) and player γ decides to play equilibrium (b, b, b), the resulting action is (a, a, b) yielding a reward of -20.

On the other hand, the actions (a, b, c), (a, c, b), (b, a, c), (b, c, a), (c, a, b) and (c, b, a) are also Nash equilibria. Even if the team uses a coordination mechanism such as fictitious play, there are no guarantees that they will cordinate in one of the optimal equilibria.

This problem is known as an *equilibrium selection problem* in the game theory literature, or as a *coordination problem* in the multi-agent systems literature [2]. Even if all players know the game, it is still necessary to devise some specific mechanism to ensure that, in the presence of multiple equilibria, all players will commit to the same equilibrium. This mechanism can rely on implicit assumptions on the way the other players play [15], communication [9], social conventions [8] or coordination graphs [10, 11, 13, 14].

In this paper we are interested in addressing coordination as a result of interaction among the agents: coordination should *emerge* from the interaction among the several players rather than being "intrinsically implanted" in the players. We also consider that no *explicit communication*

takes place. It is possible to find different works in the literature addressing the problem of emerging coordination in multi-agent systems.

Joint-action learners [7] use *fictitious play* to estimate the strategies followed by the other players in team games. This estimate on the other players' strategy is then used to choose a best response strategy. Several variations of the fictitious play principle have been proposed to ensure convergence to a coordinated equilibrium.

Adaptive play [28] is a variation of fictitious play that sub-samples the history of past-plays. This sampled-history is then used to estimate the other players' strategy in a fashion similar to fictitious player. *Biased adaptive play* [26] further extends the adaptive play strategy. The advantage of biased adaptive play over adaptive play is that the former actually converges to a coordinated equilibrium in any team Markov game, unlike the latter, whose convergence guarantees limit to which weakly acyclic repeated games.

Lauer and Riedmiller [15] propose still another strategy to ensure coordination. In this work, each player optimistically assumes that all players behave greedily. As shown in [15], this approach converges to an optimal Nash equilibrium even if the joint actions are not observable, as long as the transitions are deterministic.

1.4 Biased adaptive play

To describe how BAP works, we start with an important definition. Let $\Gamma = \left(N, (\mathcal{A}^k), r\right)$ be a fully cooperative matrix game and let D be a set containing some of the Nash equilibria in Γ (and no other joint actions). Γ is a *weakly acyclic w.r.t. the bias set* D if, given any vertex a in the best response graph of Γ, there is a directed path to either a Nash equilibrium in D or a strict Nash equilibrium.

Now, considering a fully cooperative repeated game $\Gamma = \left(N, (\mathcal{A}^k), r\right)$,[3] we construct the virtual game $VG = \left(N, (\mathcal{A}^k), r_{VG}\right)$, where $r_{VG}(a) = 1$ if a is an optimal equilibrium for Γ and $r_{VG}(a) = 0$ otherwise. Notice that every Nash equilibrium in VG corresponds to an optimal equilibrium in $\hat{\Gamma}$. Therefore, if the players are able to coordinate in a Nash equilibrium in VG, they will have coordinated in an optimal equilibrium in Γ, as desired [26]. By setting $D = \{a \in \mathcal{A} \mid r_{VG}(a) = 1\}$, the game VG is weakly acyclic w.r.t. the set D.

Let K and m be two integers such that $1 \leq K \leq m$ and let H_t be a vector with the last m joint plays at the t^{th} play of the game. We refer to any set of K samples randomly drawn from H_t without replacement as a *K-sample* and denote it as $K(H_t)$. A player k following BAP draws a K-sample $K(H_t)$ from the history of the m most recent plays and checks if

1. There is a joint action $a^* \in D$ such that, for all the actions $a \in K(H_t)$, $a^{-k} = (a^*)^{-k}$;

2. There is at least one action $a^* \in D$ such that $a^* \in K(H_t)$.

If these conditions are verified this means that, from the sample $K(H_t)$, all players (except k) appear to have coordinated in an optimal action $a^* \in D$. Therefore, if conditions 1 and 2 are met, player k chooses its best response $(a^*)^k$ such that

$$a^* = \max_{\tau \leq t} \{a_\tau \mid a_\tau \in K(H_t) \text{ and } a_\tau \in D\}.$$

If either 1 or 2 (or both) do not hold, then player k uses the K-sample to estimate the strategies of the other players as

$$EP_t^k(a^k) = \sum_{a^{-k} \in \mathcal{A}^{-k}} r(a^{-k}, a^k) \frac{N_K(a^{-k})}{K},$$

[3] A repeated game is simply a matrix game that is played repeatedly and in which the players "remember" the past plays of the game.

where $N_K(a^{-k})$ denotes the number of times that the reduced action a^{-k} appears in the K-sample $K(H_t)$. It then chooses its action randomly from the best response set

$$BR_t^k = \left\{ a^k \mid a^k = \arg\max_{b^k \in \mathcal{A}^k} EP_t(b^k) \right\}.$$

It has been shown that BAP ensures coordination w.p.1 as $t \to \infty$ as long as $m \geq K(N+2)$—see Theorems 1 and 3 and Lemma 4 in [26].

We conclude this section with one important observation, concerning the application of BAP to fully cooperative Markov games. Although these method was presented for a repeated game framework, it is possible to apply it *mutatis mutandis* to the Markov game framework. In fact, as we show next, the Q-values for an optimal Nash equilibrium define for each state of the Markov game a fully cooperative, weakly acyclic matrix game to which BAP can be applied, as long as every state x is visited infinitely often [26].

1.5 Markov games

Once again, let $\mathcal{X}$ be a finite set of states and $\{X_t\}$ a $\mathcal{X}$-valued controlled Markov chain. The transition probabilities for the chain are given by a probability function

$$\mathbb{P}\left[X_{t+1} = j \mid X_t = i, A_t = a\right] = \mathsf{P}_a(i,j),$$

where $i, j \in \mathcal{X}$ and $a \in \mathcal{A}$. As in MDPs, the $\mathcal{A}$-valued process $\{A_t\}$ represents the control process, where $\mathcal{A}$ is the finite set of possible actions. However, and unlike MDPs, each A_t is now a *joint control action* arising from N independent decision-makers (which we henceforth refer to as *players*). Therefore, A_t is a tuple $(A_t^1, \ldots, A_t^N)$, where A_t^k is the individual action for player k and takes values in the set of player k's individual actions, $\mathcal{A}^k$. The set $\mathcal{A}$ is the cartesian product of the N sets of individual actions, $\mathcal{A} = \times_{k=1}^N \mathcal{A}^k$, and is the *joint action set*. Suppose also that there is a function r defined in $\mathcal{X} \times \mathcal{A} \times \mathcal{X}$ but taking values in $\mathbb{R}^N$. It is therefore possible to write

$$r(i,a,j) = \left(r^1(i,a,j), \ldots, r^N(i,a,j)\right),$$

each r^k defined on $\mathcal{X} \times \mathcal{A} \times \mathcal{X}$ and taking values in $\mathbb{R}$.

A Markov game is thus a generalized Markov decision process $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ in which the set $\mathcal{A}$ and the reward function r are as described above. There are N independent decision-makers (the players), each choosing the value of one individual control parameter A_t^k. Each transition (i,a,j) grants player k with a reward $r^k(i,a,j$. This leads to the following definition.

Markov Game

A *Markov game* is a tuple $\left(N, \mathcal{X}, (\mathcal{A}^k), \mathsf{P}, (r^k), \gamma\right)$, where

- N is the set of players in the game;
- $\mathcal{X}$ is the set of game-states;
- $\mathcal{A} = \times_{k=1}^N \mathcal{A}^k$ is the cartesian product of the individual action sets $\mathcal{A}^k$;
- P represents the transition probability kernels. In the simplest case where $\mathcal{X}$ is finite, each P_a is a matrix with xypentry given by

$$\mathsf{P}_a(i,j) = \mathbb{P}\left[X_{t+1} = j \mid X_t = i, A_t = a\right];$$

- $r = (r^1, \ldots, r^N)$ is the reward function, assigning a reward $r^k(i,a,j)$ to player k each time a transition from i to j occurs "under" the joint action a.

In this paper, we focus on a particular class of Markov games, *team Markov games*. Team Markov games, also known as fully cooperative Markov games, arise when all players have common interests. This is translated in terms of the reward function by setting all agents to receive a *common reward*. A team Markov game is thus a tuple $\left(N, \mathcal{X}, (\mathcal{A}^k), \mathsf{P}, r, \gamma\right)$, where N is the number of players, $\mathcal{X}$ is the set of states, $\mathcal{A} = \times_{k=1}^{N} \mathcal{A}^k$ is the set of joint actions and $\mathsf{P} : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \longrightarrow \mathbb{R}$ is the transition probability function. As in MDPs, we assume that there is a deterministic function $r : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \longrightarrow \mathbb{R}$ assigning a reward $r(i,a,j)$ every time a transition from i to j occurs after taking the joint action a.

In team Markov games all players share the same goal, which is to maximize the total expected reward over all admissible control sequences $\{A_t\}$, defined as

$$V(\{A_t\}, i) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(X_t, A_t) \mid X_0 = i\right]$$

with $i \in \mathcal{X}$ and $R(i,a)$ the random reward received by *all* players for taking the joint action a in state i.

Given the game $\left(N, \mathcal{X}, (\mathcal{A}^k), \mathsf{P}, r, \gamma\right)$, the *optimal value function* V^* is defined for each state $i \in \mathcal{X}$ as

$$V^*(i) = \max_{\{A_t\}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(X_t, A_t) \mid X_0 = i\right].$$

As in MDPs, the optimal value function verifies

$$V^*(x) = \max_{a \in \mathcal{A}} \sum_{i \in \mathcal{X}} \left[r(i,a,j) + \gamma V^*(j)\right] \mathsf{P}_a(i,j).$$

and we define the optimal Q-values $Q^*(i,a)$ as

$$Q^*(i,a) = \sum_{i \in \mathcal{X}} \left[r(i,a,j) + \gamma V^*(j)\right] \mathsf{P}_a(i,j).$$

As in matrix games, an *individual strategy for player k* is a mapping $\sigma_t^k : \mathcal{X} \times \mathcal{A} \longrightarrow [0,1]$ such that the individual control sequence $\left\{A_t^k\right\}$ generated by σ_t^k verifies

$$\mathbb{P}\left[A_t^k = a^k \mid X_t = i\right] = \sigma_t^k(i,a). \tag{1.3}$$

A strategy σ_t^k is a *pure strategy* if, for each $i \in \mathcal{X}$, $\sigma^k(i,a^k) = 1$ for some action $a^k \in \mathcal{A}^k$ and a *mixed strategy* otherwise. If σ_t^k does not depend on t, it is said to be a *stationary strategy* and simply denoted by σ^k.

A *joint strategy* or *strategy profile* is a vector $\sigma_t = (\sigma_t^1, \ldots, \sigma_t^N)$ of individual strategies and defines the probability of the team playing each joint action in $\mathcal{A}$ in each state of the game. A strategy profile σ_t generates a control sequence $\{A_t\}$, where each A_t is a N-tuple $(A_t^1, \ldots, A_t^N)$ and each A_t^k verifies (1.3). We write $V^{\sigma_t}(i)$ instead of $V(\{A_t\}, i)$ whenever the control sequence $\{A_t\}$ is generated by the strategy profile σ_t, and refer to V^{σ_t} as being the *value function* associated with strategy σ_t. A strategy profile is stationary/pure if it is composed of stationary/pure individual strategies. The tuple

$$\sigma_t^{-k} = (\sigma_t^1, \ldots, \sigma_t^{k-1}, \sigma_t^{k+1}, \ldots, \sigma_t^N)$$

is a *reduced joint strategy* or simply a *reduced strategy*, and we write $\sigma_t = (\sigma_t^{-k}, \sigma_t^k)$ to indicate that the individual strategy of player k in the joint strategy σ_t is σ_t^k.

Two important remarks are now in order. First of all, if the definition of V^* and the existence of an optimal joint control strategy arise immediately from the corresponding results for MDPs, the fact that the decision process in team Markov games is distributed implies that coordination must be addressed explicitly [1].

On the other hand, we note that the function Q^* defines, at each state $x \in \mathcal{X}$, a fully cooperative strategic game $\Gamma_x = \big(N, (\mathcal{A}^k), Q^*(x, \cdot)\big)$, that we refer as a *stage game*. If the players coordinate in an optimal Nash equilibrium in each stage game Γ_x, they coordinate in an optimal Nash equilibrium for the team Markov game [2]. This means that, if Q^* is known and every state $x \in \mathcal{X}$ is visited infinitely often, each agent can keep a record of the last m plays in each state and apply BAP to each stage game. In the continuation, we address the situation in which the agent must learn Q^* while coordinating.

2 The CQL algorithm

In our description of the CQL algorithm, we consider two essential components: (1) *learning the game*; and (2) *learning to coordinate*. Learning the game consists in estimating the optimal Q-function; learning to coordinate consists in agreeing upon an optimal Nash equilibrium.

2.1 Learning the game

In CQL, each player uses the Q-learning update rule described in (1.2) to learn the optimal Q-values. Since all players receive the same reward and each player admittedly observes the actions played by the remaining players (a posteriori), all players maintain, at each time instant, a common estimate Q_t of Q^*. As long as every state-action pair (i, a) is visited infinitely often, the sequence $\{Q_t\}$ converges to Q^* with probability 1, *independently of the policy used to sample the game*. This result is standard and can be found in numerous references in the literature, *e.g.*, [17].

On the other hand, it is important to ensure sufficient *exploration* as the players coordinate. In fact, it is important to ensure sufficient visits to every state-action pair, even as the players coordinate. The use of exploration policies that become *greedy in the limit* has been shown to settle the issue of exploration vs. exploitation in a satisfactory way, for the purposes of this paper. Such policies, known as *greedy in the limit with infinite exploration* (GLIE) were thoroughly studied in [21] and their application to multi-agent scenarios in [17, 26]. Examples of GLIE policies include Boltzmann exploration with decaying temperature factor and ε-greedy policies with decaying exploration coefficient (ε). In this work, we adopt a Boltzmann-greedy policy that explores with a probability given by a Boltzmann distribution.

2.2 Learning to coordinate

In CQL, each player uses biased adaptive play to converge in behavior to an optimal Nash equilibrium. In particular, the players use BAP in each stage-game, thus coordinating in an optimal Nash equilibrium (as shown in [2]). However, the players do not know the function Q^* but rather an approximate estimate Q_t of Q^* that they must use to coordinate.

To achieve coordination, biased adaptive play considers a sequence $\{VG_t\}$ of virtual games built from the estimates Q_t by making use of the concept of ε-optimality. Each virtual game VG_t is thus defined as a tuple $VG_t = \big(N, (\mathcal{A}^k), r_t\big)$, where the payoff function is

$$r_t(a) = \begin{cases} 1 & \text{if } a \in \mathbf{opt}^{\varepsilon_t}(i); \\ 0 & \text{otherwise.} \end{cases}$$

We denoted by $\mathbf{opt}^{\varepsilon_t}(i)$ the set of ε_t-optimal actions with respect to Q_t at state i. Notice that, as $\varepsilon_t \to 0$, all suboptimal actions are eliminated from the virtual games VG_t. Therefore, to ensure that BAP is still able to coordinate using the estimates Q_t, we need only guarantee that $\varepsilon_t \to 0$ more slowly than $Q_t \to Q^*$. This statement is formally established in Lemma 2.1.

Consider a function $\rho : \mathbb{N} \longrightarrow \mathbb{R}$ such that, w.p.1,

$$\|Q_t - Q^*\| \leq K_0 \rho(t),$$

where K_0 is some positive constant. The following result is a generalization of Lemma 6 in [26].

Lemma 2.1. *For any team Markov game, let Λ_T be the event that, for $t > T$, $VG_t = VG$. If ε_t decreases monotonically to zero and*

$$\lim_{t\to\infty} \frac{\rho(t)}{\varepsilon_t} = 0,$$

then $\lim_{t\to\infty} \mathbb{P}\left[\Lambda_T\right] = 1$.

Proof The proof closely follows the proof of Lemma 6 in [26].

Let $i \in \mathcal{X}$ be some fixed state and let λ_T be the event that, for all $t > T$,

$$\max_{a\in\mathcal{A}} \left| Q_t^k(i,a) - Q^*(i,a) \right| < \frac{\varepsilon_t}{2}.$$

Since, by assumption,

$$\lim_{t\to\infty} \frac{\rho(t)}{\varepsilon_t} = 0,$$

it holds that

$$\lim_{t\to\infty} \frac{K_1\rho(t)}{\varepsilon_t} = 0,$$

for any positive constant K_1. Therefore, and since

$$\left\| Q_t^k - Q^* \right\| \leq K_0 \rho(t)$$

w.p.1, given any $\xi_0 > 0$ there is a time instant $T_0(\xi_0) > 0$ such that, for all $t > T_0$,

$$\mathbb{P}\left[\lambda_t\right] > 1 - \xi_0. \tag{2.1}$$

Take now two actions $a, b \in \mathcal{A}$ such that $a \in \mathbf{opt}(i)$ and b verifies

$$b = \arg\max_{u\notin\mathbf{opt}(i)} Q^*(i,u),$$

where we used the compact notation $\mathbf{opt}(i)$ to represent $\mathbf{opt}^0 Q^*(i,\cdot)$. Let $\xi_1 = |Q^*(i,a) - Q^*(i,b)|$. By assumption, $\varepsilon_t \to 0$ and, therefore, there is a time instant T_1 such that, for all $t > T_1$,

$$\varepsilon_t < \frac{\xi_1}{2}. \tag{2.2}$$

Let $T = \max\{T_0, T_1\}$. For all $t > T$ it holds with probability $p > 1 - \xi_0$ that, given any action $b \notin \mathbf{opt}(i)$,

$$\begin{aligned} Q_t(i,b) + \varepsilon_t &< Q^*(i,b) + \frac{\varepsilon_t}{2} + \varepsilon_t < \\ &< Q^*(i,b) + \frac{\xi_1}{2} + \frac{\xi_1}{4} \leq \\ &\leq \max_{u\in\mathcal{A}} Q^*(i,u) - \frac{\xi_1}{4} < \\ &< \max_{u\in\mathcal{A}} Q_t(i,u). \end{aligned} \tag{2.3}$$

The first inequality arises from (2.1); the second inequality arises from (2.2); the third inequality arises from the definition of δ and the last inequality arises from (2.1) once again. On the other hand, for all $t > T$ it holds with probability $p > 1 - \xi_0$ that, given any action $a \in \mathbf{opt}(i)$,

$$Q_t(i,a) + \varepsilon_t > Q^*(i,a) + \frac{\varepsilon_t}{2} > \max_{u \in \mathcal{A}} Q_t(i,u), \tag{2.4}$$

where the inequalities arise from (2.1) and from the fact that $Q^*(i,a) > Q^*(i,b)$ for any $a \in \mathbf{opt}(i)$.

Now joining (2.3) and (2.4), it holds with probability $p > 1 - \xi_0$ that, for all $t > T$,

$$\begin{aligned} Q_t(i,b) &< \max_{u \in \mathcal{A}} Q_t(i,u) - \varepsilon_t \\ Q_t(i,a) &> \max_{u \in \mathcal{A}} Q_t(i,u) - \varepsilon_t, \end{aligned}$$

for any actions $a \in \mathbf{opt}(i)$ and $b \notin \mathbf{opt}(i)$. The first expression implies that, for any $t > T$, no suboptimal action belongs to $\mathbf{opt}^{\varepsilon_t}(i)$; the second expression implies that all optimal actions do belong to $\mathbf{opt}^{\varepsilon_t}(i)$. This means that, for all $t > T$, $VG_t = VG$ with probability $p > 1 - \xi_0$ and, therefore, $\mathbb{P}[\Lambda_T] > 1 - \xi_0$. The conclusion of the theorem follows. □

We present in Figure 3 the complete CQL algorithm in pseudo-code.

The following result establishes the conditions under which CQL (coordinated Q-learning) is able to learn the optimal Q-function and coordinate in an optimal Nash equilibrium.

Theorem 2.2. *Let $\Gamma = \left(N, \mathcal{X}, (\mathcal{A}^k), \mathsf{P}, r, \gamma\right)$ be a team Markov game with N players. Suppose that the following conditions hold:*

1. *The players use the Q-learning update rule in* (1.2) *to learn the optimal Q-function;*
2. *The players use BAP with GLIE exploration to coordinate in each stage-game;*
3. *Each virtual game VG_t used in BAP considers ε_t-optimal strategies;*
4. *The sequence $\{\varepsilon_t\}$ decreases monotonically to zero and verifies*

$$\lim_{t \to \infty} \frac{\sqrt{\frac{\log\log(N_t)}{N_t}}}{\varepsilon_t} = 0, \tag{2.5}$$

   where N_t is number of visits to the least visited state-action pair at time t;

5. *The the lengths of the history H_t and K-sample h, m and K, verify $m \geq K(N+2)$;*
6. *The sequence of step-sizes $\{\alpha_t\}$ verifies*

$$\sum_{t \in \mathcal{T}} \alpha_t(i,a) = \infty; \qquad \sum_{t \in \mathcal{T}} \alpha_t^2(i,a) < \infty,$$

   and $\alpha_t(i,a) = 0$ if $(i,a) \neq (i_t, a_t)$.

Then, the sequence of estimates $\left\{Q_t^k\right\}$ generated by CQL converges to Q^* w.p.1. Furthermore, all players in N coordinate in an optimal Nash equilibrium of Γ w.p.1.

Proof Convergence of Q_t to Q^* is immediate, since the GLIE exploration ensures that every state-action pair is visited infinitely often. Since every state is visited infinitely often, we establish

Initialization:

1: **Set** $t = 0$ and $\varepsilon_t = \varepsilon_0$;

2: **For all** (i, a) **set** $n_t^k(i, a) = 1$ and $Q_t^k(i, a) = 0$;

3: **Set** $\mathbf{opt}^{\varepsilon_t}(i) = \mathcal{A}$ and $D = \mathcal{A}$;

Learning coordination: Given current state X_t

4: **If** $t \leq m$, randomly select an action

5: **else** with GLIE exploitation probability $(1 - p_t^k)$ **do**

a. **Update** VG_t^k as

$$VG_t^k(X_t, a) = \begin{cases} 1 & \text{if } a \in \mathbf{opt}^{\varepsilon_t}(X_t); \\ 0 & \text{otherwise;} \end{cases}$$

b. **Set** $D = \left\{ a \mid VG_t^k(X_t, a) = 1 \right\}$;

c. **Set** $h = K\text{-sample}(K, H_t(X_t))$;

d. **For all** $a^k \in \mathcal{A}^k$, **set**

$$EP_t^k(a^k) = \sum_{a^{-k} \in \mathcal{A}^{-k}} VG_t^k\big(X_t, (a^{-k}, a^k)\big) \frac{N_h(a^{-k})}{K};$$

e. **Set** $BR_t^k(X_t) = \left\{ a^k \mid a^k = \arg\max_{b^k \in \mathcal{A}^k} EP_t^k(a_k) \right\}$;

f. **If** conditions 1 and 2 of Section 1 are met, choose the most recent joint action in $h \cap D$;

g. **else** randomly choose an action in $BR_t^k(X_t)$;

6: **else** with exploration probability p_t^k randomly select an action;

Learning the game: Given current transition triplet (X_t, A_t, X_{t+1})

7: **Set** $n_t(X_t, A_t) = n_t(X_t, A_t) + 1$;

8: **Update** Q_t^k according to (1.2), with $\alpha_t(i, a) = \frac{1}{n_t(i,a)}$;

9: **Set** $t = t + 1$ and $N_t = \min_{i,a} n_t(i, a)$;

10: **If** $\varepsilon_t \geq \varepsilon_0 \rho_t$,

a. **Set** $\varepsilon_t = \varepsilon_0 \rho_t$;

b. **For all** i, **set** $\mathbf{opt}^{\varepsilon_t}(i) = \left\{ a \mid Q_t^k(i, a) \geq \max_b Q_t^k(i, b) - \varepsilon_t \right\}$

Figure 3: The CQL algorithm for player k.

convergence to a Nash equilibrium in Γ by establishing convergence to a Nash equilibrium in any state game $\left(N, (\mathcal{A}^k), Q^*(i,\cdot)\right)$ [2].

We start by remarking that the estimates produced by Q-learning have been shown to verify w.p.1 the following error bound [23]:

$$\max_{a\in\mathcal{A}} \left| Q_t^k(i,a) - Q^*(i,a) \right| \leq \left\| Q_t^k - Q^* \right\| \leq$$
$$\leq K_0 \sqrt{\frac{\log\log(N_t)}{N_t}},$$

for some positive constant K_0. By Lemma 2.1, for any $\xi_1 > 0$ there is $T_1(\xi_1)$ such that $VG_t = VG$ for every $t > T_1$ with probability $1 - \xi_1$. By convergence of BAP [26], if $VG_t = VG$ for all $t > T$, given any $\xi_2 > 0$, there is $T_2(\xi_2, T)$ such that all players coordinate in an optimal Nash equilibrium for the stage-game $\left(N, (\mathcal{A}^k), Q^*(i,\cdot)\right)$ for any $t > T_2$ with probability $1 - \xi_2$.

All the reasoning so far implies that there is a time instant $T_3(\xi_1, \xi_2)$ such that BAP with GLIE exploration coordinates in an optimal Nash equilibrium for the stage-game $\left(N, (\mathcal{A}^k), Q^*(i,\cdot)\right)$ with probability $(1-\xi_1)(1-\xi_2) > 1 - \xi_1 - \xi_2$. Since ξ_1 and ξ_2 are arbitrary, the conclusion of the theorem follows. □

3 Some illustrative results

We now present illustrative results obtained with the CQL algorithm in several different multi-agent problems.

The first set of tests considers the 3-player repeated game in Section 1 and analyzes the behavior of CQL against suboptimal and/or stationary teammates. We tested CQL in the game of Fig. 2, where 3 players (α, β and γ) repeatedly engage in choosing one of three possible actions a, b and c.

Figure 4 depicts the results obtained with 3 learning players in the repeated game of Fig. 2. The probability of coordination converges to 1 and, as seen in the cumulative reward plot, they coordinate to an optimal Nash equilibrium (the slope of the curve is 10).

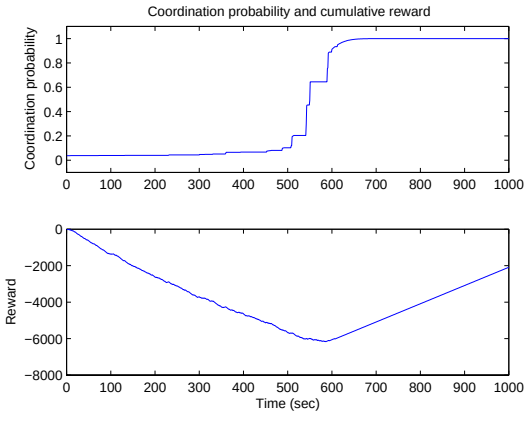

Figure 4: Probability of coordination and cumulative reward obtained in the game of Fig. 2 with 3 learning agents. Seconds represent time-steps.

We then trained CQL with two stationary teammates playing suboptimal strategies. In particular, player β always chose action a and player γ always chose action b. In this situation, as illustrated in Figure 5, CQL converges to a best response Nash equilibrium (the slope of the curve after coordination is 5).

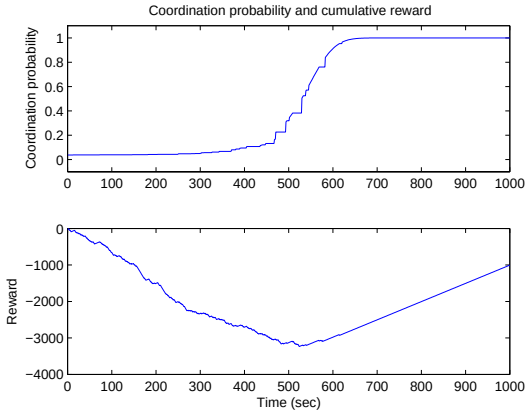

Figure 5: Results obtained in the repeated game of Fig. 2 with 2 stationary players playing, respectively, a and b.

Still in the repeated game of Fig. 2, we trained two CQL agents against one stationary teammate playing action a. After coordination, the two CQL players settle once again in an optimal Nash equilibrium, as seen in Figure 6 (the slope of the curve is, once again, 10).

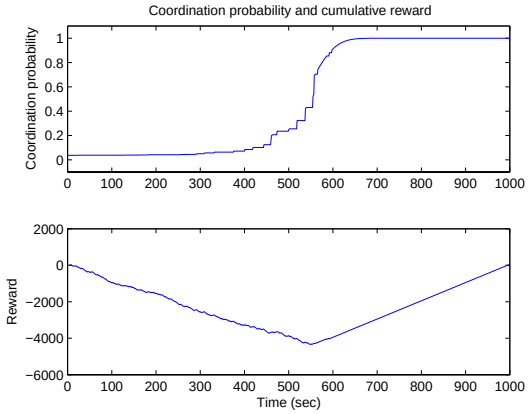

Figure 6: Results obtained in the repeated game of Fig. 2 against one stationary teammate playing a.

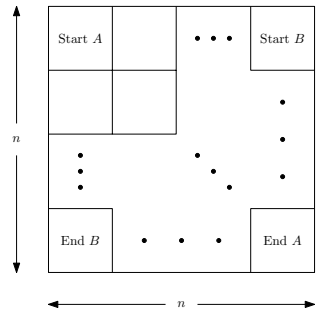

Figure 7: Generic $n \times n$ grid world.

We then tested the behavior of CQL in several small gridworld problems. We present in Figures 8 and 9 the results obtained in 2×2 and 3×3 grid worlds. In both problems, each of two players must reach the opposite corner (see Figure 7). When both players reach the corresponding corners, they receive a common reward of 20. If they "collide" in some state,

they receive a reward of -10. Otherwise, they receive a reward of 0. Each player has 4 possible actions: N, S, E and W, which makes a total of 16 possible joint actions. Each individual action moves the player in the intended direction with probability 0.9 and leaves the player's position unchanged with probability 0.1.

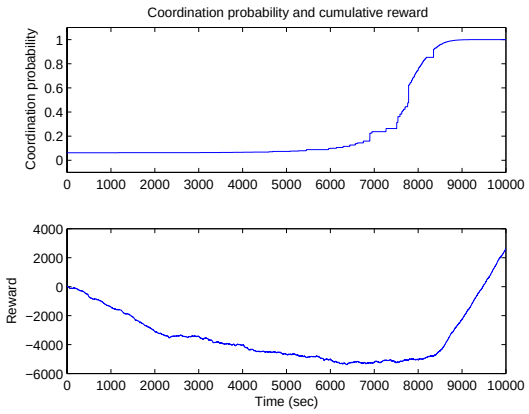

Figure 8: Probability of coordination and cumulative reward obtained in the 2×2 grid game.

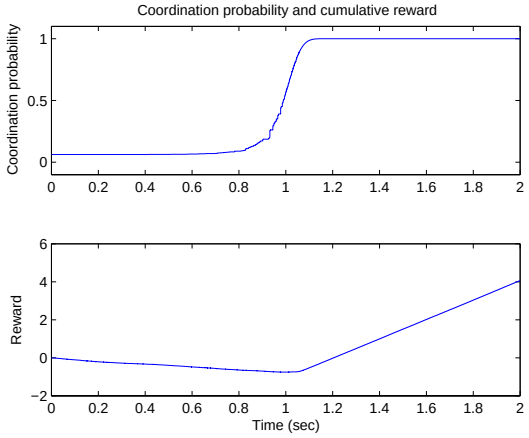

Figure 9: Probability of coordination and cumulative reward obtained in the 3×3 grid game. The values in the time scale should be multiplied by 10^5.

In both grid-world tests both players learned to coordinate using the CQL algorithm. Notice that the probability of coordination converges to 1 and, as seen in the cumulative reward plot, they are capable of coordinating in an optimal Nash equilibrium (both curves present positive slope).

As a final observation, the results obtained in the repeated matrix game illustrate (in a simple case) two important properties of CQL: convergence to a best response in the presence of stationary teammates and convergence in self-play to optimal equilibria. These two properties, labeled in [5] as *rationality* and *convergence*, hold in general by construction, as can easily be shown from Theorem 2.2.

4 Discussion

To conclude the paper, several important remarks are in order.

First of all, the CQL algorithm presented here is closely related to optimal adaptive learning (OAL) as described in [26]. While CQL combines Q-learning with biased adaptive play, OAL

combines model-based learning with biased adaptive play. The only complication in combining Q-learning with biased adaptive play resides in suitably choosing the sequence ε_t so as to verify the bound in Lemma 2.1. This difficulty is resolved by considering the rate of convergence of Q-learning analyzed in works such as [23]. In the paper we describe the combined algorithm and suitably establish convergence with probability 1 by providing the required conditions on ε_t.

A second remark is related with the problem of coordination in multi-agent learning problems. As already stated, when considering multi-agent reinforcement learning problems, coordination should always be explicitly accounted for. The existence of multiple equilibrium strategies may lead the joint behavior of a group of agents to be arbitrarily poor if no coordination is enforced, even if all agents know exactly the game they are playing. If the agents are to learn to coordinate while learning the game itself, the coordination mechanism must be supported by the past history of the game. Examples of history-based coordination mechanisms include fictitious play, adaptive play or biased adaptive play.

One third important remark is related with the use of on-policy algorithms with biased adaptive play: as we described the Q-learning update mechanism used in CQL to learn the function Q^*, one could question if an on-policy update mechanism (such as SARSA) could be used to learn the game, replacing the Q-learning update in CQL.

Alas, the answer to this question is negative. In fact, SARSA converges to the optimal Q-function only if a GLIE strategy is used for learning; if any other policy is used, SARSA converges to the corresponding Q-values. In CQL, the bounds on ε_t imply that coordination occurs only when the estimates Q_t are "sufficiently close" to the true function Q^*. Using a SARSA-like update, the estimates Q_t approach Q^* only as the learning strategy approximates the greedy strategy (while still ensuring sufficient exploration). These are two incompatible requirements and, therefore, it is not generally possible to use an on-policy update mechanism such as SARSA with BAP.

Future work addressing complex environments (with large/infinite state-spaces) should take into account the impact of compact representations of the state-space on how coordination can now be obtained from the history of the process.

Acknowledgements

The authors would like to acknowledge the helpful discussions with Prof. Manuela Veloso from CMU.

References

[1] Craig Boutilier. Sequential optimality and coordination in multiagent systems. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 478–485, 1999.

[2] Craig Boutilier. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-96)*, pages 195–210, 1996.

[3] Michael Bowling and Manuela Veloso. An analysis of stochastic game theory for multiagent reinforcement learning. Technical Report CMU-CS-00-165, School of Computer Science, Carnegie Mellon University, 2000.

[4] Michael Bowling and Manuela Veloso. Scalable learning in stochastic games. In *Proceedings of the AAAI Workshop on Game Theoretic and Decision Theoretic Agents (GTDT'02)*, pages 11–18. The AAAI Press, 2000. Published as AAAI Technical Report WS-02-06.

[5] Michael Bowling and Manuela Veloso. Multi-agent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.

[6] George W. Brown. Some notes on computation of games solutions. Research Memoranda RM-125-PR, RAND Corporation, Santa Monica, California, 1949.

[7] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI'98)*, pages 746–752, 1998.

[8] Nicholas V. Findler and Raphael M. Malyankar. Social structures and the problem of coordination in intelligent agent societies. Invited talk at the special session on "Agent-Based Simulation, Planning and Control", IMACS World Congress (2000), 2000. CD Paper 122-12.

[9] Felix Fischer, Michael Rovatsos, and Gerhard Weiss. Hierarchical reinforcement learning in communication-mediated multiagent coordination. In C. Sierra and L. Sonenberg, editors, *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, pages 1334–1335, 2004.

[10] Carlos Guestrin, Daphne Koller, and Ronald Parr. Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems*, volume 14, pages 1523–1530, 2001.

[11] Carlos Guestrin, Michail G. Lagoudakis, and Ronald Parr. Coordinated reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML'02)*, pages 227–234, 2002.

[12] Junling Hu and Michael P. Wellman. Nash *Q*-learning for general sum stochastic games. *Journal of Machine Learning Research*, 4:1039–1069, 2003.

[13] Jelle R. Kok, Matthijs T. J. Spaan, and Nikos Vlassis. An approach to noncommunicative multiagent coordination in continuous domains. In Marco Wiering, editor, *Benelearn 2002: Proceedings of the Twelfth Belgian-Dutch Conference on Machine Learning*, pages 46–52, Utrecht, The Netherlands, 2002.

[14] Jelle R. Kok, Matthijs T. J. Spaan, and Nikos Vlassis. Multi-robot decision making using coordination graphs. In *Proceedings of the 11th International Conference on Advanced Robotics (ICAR'03)*, pages 1124–1129, Coimbra, Portugal, 2003.

[15] Martin Lauer and Martin Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proceedings of the 17th International Conference on Machine Learning (ICML'00)*, pages 535–542, San Francisco, CA, 2000. Morgan Kaufmann.

[16] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In Ramon López de Mántaras and David Poole, editors, *Proceedings of the 11th International Conference on Machine Learning (ICML'94)*, pages 157–163, San Francisco, CA, 1994. Morgan Kaufmann Publishers.

[17] Michael L. Littman. Value-function reinforcement learning in Markov games. *Journal of Cognitive Systems Research*, 2(1):55–66, 2001.

[18] Michael L. Littman. Friend-or-foe *Q*-learning in general-sum games. In *Proceedings of the 18th International Conference on Machine Learning (ICML'01)*, pages 322–328, San Francisco, CA, 2001. Morgan Kaufmann Publishers.

[19] Don Monderer and Lloyd S. Shapley. Fictitious play property for games with identical interests. *Journal of Economic Theory*, 68:258–265, 1996.

[20] Sandip Sen and Gerhard Weiß. *Learning in multiagent systems*, chapter 6, pages 259–298. The MIT Press, 1999.

[21] Satinder P. Singh, Tommi Jaakkola, Michael Littman, and Csaba Szepesvari. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38(3):287–310, 2000.

[22] Satinder P. Singh, Michael Kearns, and Yishay Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI'00)*, pages 541–548, 2000.

[23] Csaba Szepesvári. The asymptotic convergence rates for Q-learning. In *Proceedings of Neural Information Processing Systems (NIPS'97)*, volume 10, pages 1064–1070, 1997.

[24] Csaba Szepesvári and Michael L. Littman. A unified analysis of value-function-based reinforcement learning algorithms. *Neural Computation*, 11(8):2017–2059, 1999.

[25] William Uther and Manuela Veloso. Adversarial reinforcement learning. Technical Report CMU-CS-03-107, School of Computer Science, Carnegie Mellon University, January 2003.

[26] Xiaofeng Wang and Tuomas Sandholm. Reinforcement learning to play an optimal Nash equilibrium in team Markov games. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 1571–1578. MIT Press, Cambridge, MA, 2003.

[27] Christopher Watkins and Peter Dayan. Technical note: Q-learning. *Machine Learning*, 8:279–292, 1992.

[28] H. Peyton Young. The evolution of conventions. *Econometrica*, 61(1):57–84, January 1993.